PATSEEK:
A Content Based Image Retrieval System For Patent Database

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Technology

by

Avinash Tiwari

To the

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERIN

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

July, 2004

CERTIFICATE

It is certified that the work contained in this thesis entitled "*PATSEEK: A Content Based Image Retrieval System For Patent Database*" has been carried out by **Mr. Avinash Tiwari (Roll No. Y211405)** under my supervision and the work has not been submitted elsewhere for a degree.

Veena Bansal

July, 2004

Dr. Veena Bansal
Assistant Professor
Industrial and Management Engineering
Indian Institute of Technology
Kanpur-208016

ABSTRACT

A patent is a legal agreement between a country and an inventor giving the inventor the right to exclude others from making, using, or selling an invention for a limited time in that country. Researchers in the sciences often have need of patent literature as an information source. This data is used to promote new directions in research, for new uses for existing technologies and to predict growth industries. In addition, patents can be the sole source of technical information on a particular invention or process. The patent information has been growing at an enormous pace and due to the amount of information available in these repositories; it became infeasible for the human beings to manually retrieve the required information. As the information has been digitized, the text based search systems were developed to cater the needs.

The patent repositories have two important components, namely, the text and the image. The research in the past has been centered on the text content of the databases. In the present work, we have developed a prototype image retrieval system which utilizes the drawings in the patent databases. The developed system automatically creates a local image database based on the keywords supplied by the user. The local database is created by searching the United States Patent Database (USPTO) patent database. The system provides a user interface to search the local database using query-by-image. As an output, the system provides top twelve images which are most similar to the query image in the database. The shape based image representation, edge orientation auto-correlogram (EOAC) has been used as the image feature representation and the recall rate has been cent percent for sixty-one percent of the queries.

ACKNOWLEDGEMENT

I would like to express deep gratitude to my thesis supervisor Dr. Veena Bansal for her invaluable guidance and encouragement which inspired me throughout. She continuously encouraged me through her valuable suggestions and constructive criticism. Without her it would had been impossible for me to come this far.

I would also like to thank my parents for the constant support, appreciation and priding on me.

The life at IIT Kanpur would not have been so memorable and enjoying without the company of my dearest friends and batch-mates. Last but not the least, thanks are due to the faculty and staff of IME for accepting me as a family member.

July, 2004 Avinash Tiwari

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Twentieth century has witnessed unparalleled growth in the number, availability and importance of images in all walks of life. Images now play a crucial role in fields as diverse as medicine, journalism, advertising, design, education and entertainment. Technology, in the form of inventions such as photography and television, has played a major role in facilitating the capture and communication of images. But the real engine of the imaging revolution has been the computer, bringing with it a range of techniques for digital image capturing, processing, storage and transmission.The creation of the World-Wide Web in the early 1990s, enabling users to access data in a variety of media from anywhere on the planet, has provided a further massive stimulus to the exploitation of digital images. Tera bytes of data are been generated in the form of aerial imagery, surveillance images, fingerprints, trademarks and logos, graphic illustrations, engineering line drawings, documents, manuals, medical images. These large repositories of images needs techniques to find desired images according to some specified criteria.

Initial text-based approaches were based on associating textual information, like filename, captions and keywords, for every image stored in the repository[1]. In text-based image retrieval the images were annotated based on their content, and these annotations were stored in traditional database. For image retrieval, keyword based matching was employed for finding the relevant images. Two major limitations hindered the growth of text-based image retrieval. Firstly for the manual annotation of vast amount of images present in the repository, the labor requirement was prohibitive. The second limitation results from the difficulty in capturing the rich content of images using a small number of key words, a problem which is compounded by the subjectivity of human perception. For example, a query for all the images in the repository with "people" in it will give good results if we annotate all the images containing people, but for the same annotations, a specific search for images with men or women in it will fail.

Image retrieval based on image content is more desirable and more effective in a number of applications. Though it seems effortless for a human being to pick out photos of horses from a small collection of pictures, the very size of image repositories makes it infeasible for him/her to find relevant images from it. As a result, there is a need to automatically extract primitive visual features from the images and to retrieve images on the basis of these features. This lead to the development of Content-Based Image Retrieval(CBIR). The earliest use of the term content-based image retrieval in the literature seems to have been by Kato [2], to describe his experiments into automatic retrieval of images from a database by colour and shape feature. Content-based image retrieval (CBIR) is aimed at efficient retrieval of relevant images from large image databases based on automatically derived image features. These features are typically extracted from shape, texture, or color properties of query image and images in the database. Potential applications include digital libraries, commerce, Web searching, geographic information systems, biomedicine, surveillance and sensor systems, education, crime prevention, etc.

1.1 Content Based Image Retrieval

A typical content-based image retrieval (CBIR) system is depicted in Fig.1.1. The image collection database contains raw images for the purpose of visual display. The visual feature repository stores visual features extracted from images needed to support content-based image retrieval. The text annotation repository contains key words and free-text descriptions of images. Multidimensional indexing is used to achieve fast retrieval and to make the system scalable to large image collections.

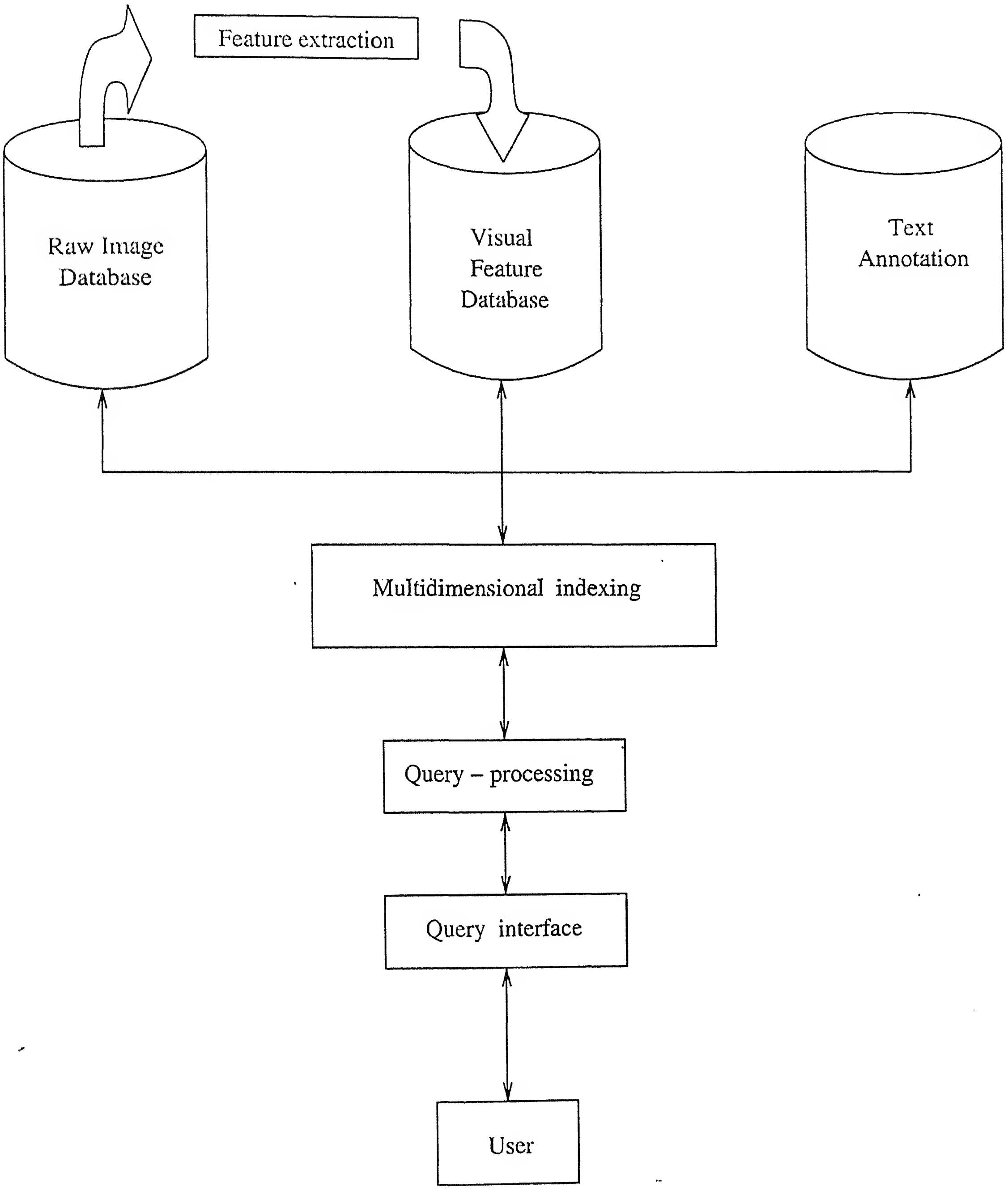


Figure 1.1: An content based image retrieval system architecture.

The retrieval engine includes a query interface and a query-processing unit. The query interface, typically employing graphical displays and direct manipulation techniques, collects information from users and display the retrieval results. The query-processing unit is to translate user queries into an internal form, which is then submitted to the DBMS. Moreover, in order to gap the bridge between low-level visual features and high-level semantic meanings, users are usually allowed to communicate with search engine in an interactive way.

1.2 Feature Extraction and Integration

Feature extraction is the basis of CBIR. Features can be categorized as general or domain-specific. General features typically include color, texture, shape, sketch, spatial relationships, and deformation, whereas domain-specific features like human faces and finger prints are applicable in specialized domains such as human face recognition and fingerprint recognition respectively.

Each feature may have several representations. For example, color histogram and color moments are both representations of the image color feature. Moreover, numerous variations of the color histogram itself have been proposed, each of which differs in the selected color-quantization scheme.

1.3 Feature Based Representation

Here, we discusses the primitive features color, texture and shape.

1.3.1 Color

The color feature is probably the most visible feature for most humans. It has been widely used in image retrieval systems like QBIC [4] and Virage [5]. An added benefit of using color features is that they are invariant to image scaling, translation, and rotations these transformations does not effect the color content of the image. The key issues in color feature extraction include the color space, color quantization, and the choice of similarity function.

Color Spaces Before an image can be indexed in a CBIR system, a proper transformation to a suitable color space is required. A color space is defined as a model for representing color in terms of intensity values. A color space defines a one- to four-dimensional space. A color component, or a color channel, is one of the dimensions. A one-dimensional space (i.e., one dimension per pixel) represents the gray-scale space.

Color spaces are related to each other by very simple mathematical formulas. The following is a list of commonly used color spaces in image processing and image indexing :

- *Gray spaces*

  Gray spaces typically have one single component, ranging from black to white. Gray spaces are the most common color space in biomedical imaging, as most medical scanners produce 2-D or 3-D (spatially) gray-scale images and 2-D electrophoresis gels are typically of gray-scale.

- *RGB-based spaces*

  The RGB space is a three-dimensional color space with components representing the red, green, and blue intensities that make up a given color. The RGB-based spaces are commonly used for devices such as color scanners and color monitors. They are also the primary color spaces in computer graphics due to the hardware support. The family of the RGB-based spaces include the RGB space, the HSV (hue, saturation, value) space, and the HLS (hue, lightness, saturation) space. Any color expressed in the RGB space is a mixture of three primary colors: red, green, and blue. For example, the color cyan can be viewed as the combination of the blue color and the green color. The HSV space and the HLS space are transformations of RGB space that can describe colors in terms more natural to a person. The HSV and the HLS spaces are slightly different in their mathematical definitions.

- *CMYK-based spaces*

  CMYK stands for Cyan, Magenta, Yellow, and blacK. CMYK-based color spaces model the way dyes or inks are applied to paper in the printing or drawing process. Ideally, the relation between RGB values and CMYK values is as simple as:

$$\begin{aligned} C &= max - R \\ M &= max - G \\ Y &= max - B \\ K &= 1 \ when \qquad R = G = B = 0 \\ K &= 0 \ when \ R \neq 0, G \neq 0, or B \neq 0 \end{aligned} \tag{1.1}$$

Here max is the maximum possible value for each color component in the RGB color space. For a standard 24-bit color image, max = 255.

- *CIE based spaces*

  The RGB color spaces and the CMYK color spaces are all device-dependent because they were developed mainly to facilitate computer devices including monitors and printers. They are not very well correlated to the human perception. There are classes of color spaces that can express color in a device-independent way. They are based on the research work done in 1931 by the Commission Internationale d'Eclairage (CIE). They are also called interchange color spaces because they are used to convert color information from the native color space of one device to the native color space of another device. XYZ, CIE LUV, CIE Lab are examples of the CIE-based color spaces.

  The CIE-based color spaces simulates human color perception. Research in human vision has revealed that three sensations are generated after the sensory membrane in the eye (or the retina) receives three color stimuli (red, green and blue). The three sensations are a red-green sensation, a yellow-blue sensation, and a brightness sensation. The CIE-based color spaces are considered a global color reference systems because of its perception correlation properties.

Color Quantization Color quantization is used to reduce the color resolution of an image. Using a quantized color map can considerably decrease the computational complexity during image retrieval. The commonly used color-quantization schemes include uniform quantization, vector quantization, and tree-structured vector quantization.

- *Uniform Quantization* In uniform quantization each axis of the color space is treated independently. Each axis is then divided into equal sized segments. The planes perpendicular to the axis' that pass through the division points then define regions in the color space [27].
- *Vector Quantization* Vector quantization is the problem of selecting K vectors in some N dimensional space to represent N vectors from that space where K < N and the total error incurred by the quantization is minimized[27].
- *Tree-Structured Vector Quantization* The idea behind tree-structured vector quantization is to build a tree structure containing always a maximum of K different colors. If a further color is to be added to the tree structure, its color value has to be merged with the most likely one that is already in the tree. The both values are substituted by their mean[26].

Similarity Functions A similarity function is a mapping between pairs of feature vectors and a positive real-valued number, which is chosen to be representative of the visual similarity between two images. For example, there are two main approaches to color histogram formation. The first one is based on the global color distribution across the entire image, whereas the second one consists of computing the local color distribution for a certain partition of the image. These two techniques are suitable for different types of queries. If users are concerned only with the overall colors and their amounts, regardless of their spatial arrangement in the image, then indexing using the global color distribution is useful. However, if users also want to take into consideration the positional arrangements of colors, the local color histogram will be better choice.

1.3.2 Texture

Texture can be defined as the set of local neighbourhood properties of the gray level of an image region[24]. Texture refers to visual patterns with properties of homogeneity that do not result from the presence of only a single color or intensity. Tree barks, clouds, water, bricks, and fabrics are example of texture. Typical texture features include contrast, uniformity, coarseness, roughness, frequency, density, and directionality. Texture features usually contain important information about the structural arrangement of surfaces and their relationship to the surrounding environment. Different classes of methods for texture extraction are[25]:

- *Statistical methods*: These methods gather information about textures by exploiting pixel statistics. The statistics can be first-order statistics like histogram mean and variance or higher order statistics. The most commonly used methods are based on the gray-level co-occurrence matrix, from which texture features are derived. A co-occurrence matrix counts how often pairs of gray levels of pixels, separated by a certain distance and lying along certain direction, occur in a image.

- *Model based methods*: They contruct a generative or stochastic model of textures. The parameters of the model are estimated for an image and act as the feature descriptors. Successful in many applications are "random field models" such as autoregressive models and Markov random fields. An autoregressive model is a random process model in which the current value of the output is expressed as the sum of its mean value, the current values of a white noise process, and a linear aggregate of the gray values of local neighbourhood pixels. Fractal based modeling is also being used.

- *Structural methods*: They describe textures as composed of well defined texture primitives (texels), which are placed according to some syntatic rules. Since this allows only the description of very regular textures, the rules are often extended to become statistical, which offers more freedom in the description.

- *Transform methods*: They represent an image in a new form, in which the characteristics of the texture become more easily accessible. Examples of this are spectral methods, where spatial frequency information becomes clean in Fourier transformed images. An important subclass is formed by the *multiplication methods* that transform images into a new representation which separates features of different scales of resolution. Examples of this are scale space, Gabor and wavelet decomposition methods.

1.3.3 Shape

Two major steps are involved in shape feature extraction. They are object segmentation and shape representation.

Object Segmentation Image retrieval based on object shape is considered to be one of the most difficult aspects of content-based image retrieval because of difficulties in low-level image segmentation and the variety of ways a given three-dimensional object can be projected into two-dimensional shape. Several segmentation techniques have been proposed so far and include the global threshold-based technique, the region-growing technique, the split-and-merge technique, the edge-detection-based technique, the texture-based technique, the color-based technique, and the model-based technique. Generally speaking, it is difficult to do a precise segmentation owing to the complexity of the individual object shape, the existence of shadows, noise, and so on.

Shape Representation. Once objects are segmented, their shape features can be represented and indexed. In general, shape representation can be classified into three categories:

- *Boundary-Based representation (Based on the Outer Boundary of the Shape.)* The commonly used descriptors of this class include the chain code, the Fourier descriptor, and the UNL descriptor.
- *Region-Based Representations (Based on the Entire Shape Region.)* Descriptors of this class include moment invariants, Zernike moments, the morphological descriptors, and the pseudo-Zernike moments.
- *Combined Representations.* We may consider the integration of several basic representations such as moment invariants with Fourier descriptor or moment invariants with the UNL

descriptor.

1.3.4 Feature Integration

Feature integration is a strategy to potentially improve image retrieval, both in terms of speed and quality of results, by combine multiple heterogeneous features. We can categorize feature integration as either sequential or parallel. Sequential feature integration, also called *feature filtering*, is a multistage process in which different features are sequentially used to prune a candidate image set. In the parallel feature-integration approach, several features are used concurrently in the retrieval process. In the latter case, different weights need to be assigned appropriately to different features, because different features have different discriminating powers, depending on the application and specific task. The feature-integration approach appears to be superior to using individual features and, as a consequence, is implemented in most CBIR systems. The original Query by Image Content (QBIC) system [4] allowed the user to select the relative importance of color, texture, and shape. The virage system [5] allows queries to be built by combining color, composition (color layout), texture, and structure (object boundary information).

1.4 Similarity Measures and Indexing Schemes

Given a feature and its representation associated with each image, we need a metric to compare an image I of the database and the query Q. A basic way is to use a distance D. A distance is defined as follows:

$$D : \Im x \Im \rightarrow \Re^{+} \qquad (1.2)$$

where $\Im$ is the set of images and $\Re^+$ the set of positive real numbers. D must satisfy the following properties for all the images I, J and K in $\Im$.

$$\begin{array}{llcll} P_1: & D(I,I) & = & D(J,J) & self-similarity \\ P_2: & D(I,J) & \geq & D(I,I) & minimality \\ P_3: & D(I,J) & = & D(J,I) & symmetry \\ P_4: & D(I,K)+D(K,J) & \geq & D(I,J) & triangular\ inequaity \end{array} \tag{1.3}$$

Any application satisfying P_1, P_2, and P_4 is a *metric*. Any application satisfying P_1, P_2, and P_3 is a *(di)similarity*.

SIMILARITY/DISTANCE MEASURES

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly. In this section, we will introduce some commonly used similarity measures. We denote D(I, J) as the distance measure between the query image I and the image J in the database; and f_i(I) as the number of pixels in bin i of I .

Minkowski-Form Distance

If each dimension of image feature vector is independent of each other and is of equal importance, the Minkowski-form distance L_p is appropriate for calculating the distance between two images. This distance is defined as:

$$\mathbf{D(I,J)} = (\sum_i |f_i(I) - f_i(J)|^p)^{\frac{1}{p}} \tag{1.4}$$

when p=1, 2, and ∞, D(I, J) is the L_1, L_2 (also called Euclidean distance), and L distance respectively. Minkowski-form distance is the most widely used metric for image retrieval. For instance, MARS system [7] used Euclidean distance to compute the similarity between texture features;

Blobworld [8] used Euclidean distance for texture and shape feature. In addition, Voorhees and Poggio [9] used L_∞ distance to compute the similarity between texture images.

The Histogram intersection can be taken as a special case of L_1 distance, which is used by Swain and Ballard [10] to compute the similarity between color images. The intersection of the two histograms of I and J is defined as:

$$\mathbf{S(I,J)} = \frac{\sum_{t=1}^{N} min(f_t(I), f_t(J))}{\sum_{t=1}^{N} f_t(J)} \tag{1.5}$$

It has been shown that histogram intersection is fairly insensitive to changes in image resolution, size, occlusion, depth, and viewing point.

Quadratic Form (QF) Distance

The Minkowski distance treats all bins of the feature histogram entirely independently and does not account for the fact that certain pairs of bins correspond to features which are perceptually more similar than other pairs. To solve this problem, quadratic form distance is introduced:

$$\mathbf{D(I,J)} = \sqrt{(\mathbf{F}_I - \mathbf{F}_J)^T \mathbf{A} (\mathbf{F}_I - \mathbf{F}_J)} \tag{1.6}$$

where A=[a_{ij}] is a similarity matrix, and a_{ij} denotes the similarity between bin i and j. F_I and F_J are vectors that list all the entries in $f_i(I)$ and $f_i(J)$.

Quadratic form distance has been used in many retrieval systems [11, 12] for color histogram-based image retrieval. It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross similarity between colors.

Mahalanobis Distance

The Mahalanobis distance metric is appropriate when each dimension of image feature vector is dependent on each other and is of different importance. It is defined as:

$$\mathbf{D(I,J)} = \sqrt{(\mathbf{F}_I - \mathbf{F}_J)^T \mathbf{C}^{-1} (\mathbf{F}_I - \mathbf{F}_J)} \tag{1.7}$$

where C is the covariance matrix of the feature vectors.

The Mahalanobis distance can be simplified if feature dimensions are independent. In this case, only a variance of each feature component, c_i, is needed.

$$\mathbf{D(I,J)} = \sum_{t=1}^{N} (\mathbf{F}^I - \mathbf{F}^J)^2 / c_t \tag{1.8}$$

Kullback-Leibler (KL) Divergence and Jeffrey-Divergence (JD)

The Kullback-Leibler (KL) divergence measures how compact one feature distribution can be coded using the other one as the codebook. The KL divergence between two images I and J is defined as:

$$\mathbf{D(I,J)} = \sum_t f_t(I) log \frac{f_t(I)}{f_t(J)} \tag{1.9}$$

The KL divergence is used in [13]as the similarity measure for texture.

The Jeffrey-divergence (JD) is defined by:

$$\mathbf{D(I,J)} = \sum_t f_t(I) log \frac{f_t(I)}{\hat{f}_t} + f_t(J) log \frac{f_t(J)}{\hat{f}_t} \tag{1.10}$$

where $\hat{f} = [f_i(I) + f_i(J)]/2$. In contrast to KL-divergence, JD is symmetric and numerically more stable when comparing two empirical distributions.

INDEXING SCHEME

Another important issue in content-based image retrieval is effective indexing and fast searching of images based on visual features. Because the feature vectors of images tend to have high dimensionality and therefore are not well suited to traditional indexing structures, *dimension reduction* is usually used before setting up an efficient indexing scheme.

One of the techniques commonly used for dimension reduction is principal component analysis (PCA). It is an optimal technique that linearly maps input data to a coordinate space such that the axes are aligned to reflect the maximum variations in the data. The QBIC system uses PCA to reduce a 20-dimensional shape feature vector to two or three dimensions [12, 15]. In addition to PCA, many researchers have used Karhunen-Loeve (KL) transform to reduce the dimensions of the feature space. Although the KL transform has some useful properties such as the ability to locate the most important sub-space, the feature properties that are important for identifying the pattern similarity may be destroyed during blind dimensionality reduction [14]. Apart from PCA and KL transformation, neural network has also been demonstrated to be a useful tool for dimension reduction of features [16].

After dimension reduction, the multi-dimensional data are indexed. A number of approaches have been proposed for this purpose, including *R-tree* (particularly, *R*-tree* [17]), *linear quad-trees* [18], *K-d-B tree*[19] and *grid files* [20]. Most of these multi-dimensional indexing methods have reasonable performance for a small number of dimensions (up to 20), but explore exponentially with the increasing of the dimensionality and eventually reduce to sequential searching. Furthermore, these indexing schemes assume that the underlying feature comparison is based on the Euclidean distance, which is not necessarily true for many image retrieval applications. One attempt to solve the indexing problems is to use hierarchical indexing scheme based on the *Self-Organization Map (SOM)* proposed in [21]. In addition to benefiting indexing, SOM provides users a useful tool to browse the representative images of each type.

1.5 User Interaction

For content-based image retrieval, user interaction with the retrieval system is crucial since flexible formation and modification of queries can only be obtained by involving the user in the retrieval procedure. User interfaces in image retrieval systems typically consist of a query formulation part and a result presentation part.

QUERY SPECIFICATION

Specifying what kind of images a user wishes to retrieve from the database can be done in many ways. The search methods used for image databases differ from those of traditional databases. Exact queries are only of moderate interest and, when they apply, are usually based on metadata managed by a traditional database management system (DBMS).The quintessential query method for multimedia databases is *retrieval-by-similarity*. The user sketch, expressed through one of a number of possible user interfaces, is translated into a query on the feature table or tables. Similarity queries are grouped into three main classes[22]:

1. *Range Search.* Find all images in which feature 1 is within range r_1, feature 2 is within range r_2, and ..., and feature n is within range r_n. Example:Find all images showing a tumor of size between size$_{min}$ and size$_{max}$ within a given region.
2. *k-Nearest Neighbor Search.* Find the k most similar images to the template. Example: Find the 20 tumors that are most similar to a specified example, in which similarity is defined in terms of location, shape, and size, and return the corresponding images.
3. *Within-Distance or (α-cut).* Find all images with a similarity score better than α with respect to a template, or find all images at distance less than *d* from a template. Example: Find all the images containing tumors with similarity scores larger then α_0 with respect to an example provided.

Note that nearest-neighbor queries are required to return at least k results, possible more in case of ties, no matter how similar the results are to the query, whereas within-distance queries do not have an upper bound on the number of returned results but are allowed to return an empty set. A query of type 1 requires a complex interface or a complex query language such as SQL. Queries of type 2 and 3 can, in their simplest incarnations be expressed through the use of simple, intuitive interface that support query-by-example.

Nearest-neighbor queries rely on the definition of a *similarity function*. These search problems have wide applicability beyond information retrieval and GIS data management. α-Cut queries rely on a distance or *scoring function*. A scoring function is nonnegative and bounded from above and assigns higher values to better matches. For example, a scoring function might order the database records by how well they match the query and then use the record rank as the score. The last record, which is the one that satisfies the query, has the highest score. Scoring functions are commonly normalized between zero and one.

1.6 Overview of the Thesis

The digital images are been generated in tremendous rate. This has led to the emergence of many image repositories containing thousands of images. One such image repository is the patent databases. For example, the Web Patent Full-Text Database (PatFT) of United States Patent Office (USPTO) contains the full-text of over 3,000,000 patents from 1976 to the present and it provides links to the Web Patent Full-Page Images Database (PatImg), which contains over 70,000,000 images, including every page of over 7,000,000 patents from 1790 to the most recent issue week [23]. The seer size of such repository makes it prohibitive for humans to find similar images in them. This has motivated us to develop a content-based image retrieval system for the patent databases. The type of images stored in these database lacks the visual features like colour and texture which have been extensively used in content-based image retrieval systems developed so far. Keeping in mind the type of images and the need of professional patent searchers, we have made an effort to automatize the processes of creation of image feature database from one such database, USPTO patent database, and develop a user interface that facilities the image retrieval process for the end-user.

Followed by the introduction of this research work, Chapter 2 starts with a review of related work on shape based feature representation of images and work done in patent search and mining. Region based and boundary-based feature representations are reviewed. The interest of researchers in techniques for searching and retrieving information from patent databases has also been reviewed.

In Chapter 3, a full-scale prototype system, PATSEEK, is introduced. PATSEEK can perform the processes of an automatic database creation to be used later for image retrieval. PATSEEK architecture is based on interactive retrieval model, which gives user a easy control over the system. PATSEEK provides the user interfaces for populating the image database according to the keywords-based search specified by the user and image retrieval by query-by-image example.

Next in Chapter 4, the overall performance of PATSEEK is evaluated. The evaluation focused on examining the effectiveness and efficiency of the system.

Finally, Chapter 5 summarizes the overall achievements of the work, and the basic assumptions of the system are reviewed. The contributions of this research work to current knowledge on patent database retrieval are addressed. Finally, some possible future research directions for extending the related techniques as well as further development of PATSEEK are presented.

Chapter 2

LITERATURE REVIEW

Shape analysis methods usually require an image to be represented as object regions or boundaries. Depending on the application, this can be achieved through some form of thresholding or edge detection. Separating objects from the background in an image is not trivial task. Objects within an image may be occluded, or boundaries may be gradational. As a consequence, not all types of data are suited to this type of representation. Both region-based and boundary-based method are investigated here.

2.1 Region based methods

In region based techniques, all the pixels within a shape region are taken into account to obtain the shape representation.

2.1.1 Scalar Geometric Parameters

The geometric attributes of a closed binary region can be used to quantify the shape of an image object. Descriptors designed to be visually meaningful are extracted from a region using a set theory approach. The parametes used to calculate some commonly used descriptors are illutrated in Fig.2.1 Formulas for the descriptors are given by:

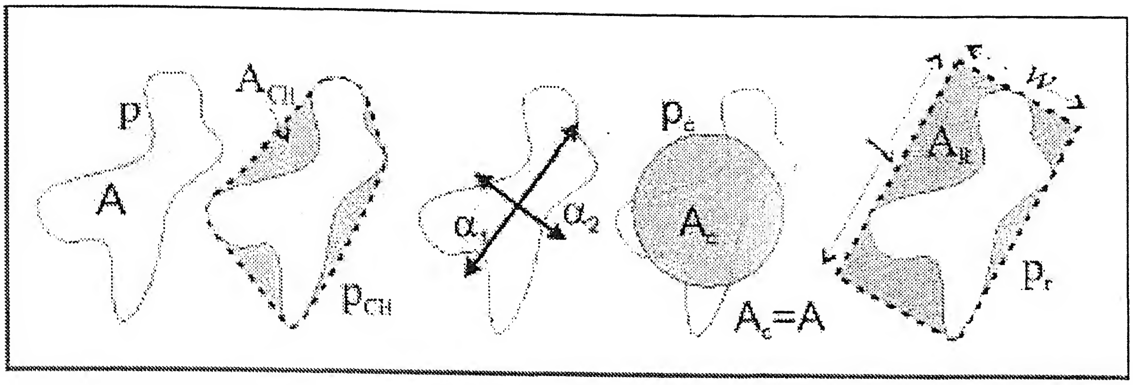


Figure 2.1: Scalar region descriptors. p - the perimeter of the region (the number of pixels that lie on the boundary of the region). A - the area of the region (the number of pixels within the region). p_{CH} - the perimeter of the convex hull of the region. A_{CH} - the area of the convex hull of the region. α_1 and α_2 - the major and minor axes of the region repectively. p_C - the perimeter of a circle with the same area as the region, that is A_C = A. p_r - the perimeter of the region bounding rectangle. A_r - the area of the region bounding rectangle. l - the length of the region bounding rectangle. w - the width of the region bounding rectangle.

• compactness	p_c/p
• principal axis ratio	α_1/α_2
• convexity	p_{CH}/p
• rectangularity	A/A_R
• elongatedness	l/w

There are many scalar geometric descriptors that have been used to characterize the shape of image objects. However,as these descriptors can only discriminate shapes with large differences, they are usually used as filters to eliminate false hits or combined with other shape descriptors to discriminate shapes. They are not suitable for standalone shape descriptors.

2.1.2 Invariant Moments

The thoery of moments provides a way of representing shapes of image objects. Moments were first used for shape description by Hu [28], who showed that moment-based shape description is information preserving.

Let f(x,y) $\geq$ 0 be a real bounded function with support on a finite region S $\in \Re^2$. The two-dimensional cartesian moment $m_{p,q}$ of order of (p+q) of the function f(x,y) is defined as:

$$m_{p,q} = \int\int_S x^p y^q f(x,y) dxdy, p,q = 0,1,2,... \tag{2.1}$$

Setting f(x,y) = 1 gives the moments of the region S that could represent a shape. For the discrete function f(i,j), the moments are computed as :

$$m_{p,q} = \sum_{(i,j)\in S} i^p j^q f(i,j) \tag{2.2}$$

The infinite set of moments $m_{p,q}$,p,q = 0,1,2, ... uniquely determines *f(x,y)* and vice-versa. Certain functions of moments are invariant to geometric transformations such as translation, scaling and rotation. Such features are useful in identification of objects with unique shapes regardless of their location, size and orientation.

1. *Translation* Under a translation of coordinates, $\acute{x}$ = x + α, $\acute{y}$ = y + β, the central moments:

$$\mu_{p,q} = \int\int_s (x-\bar{x})^p (y-\bar{y})^q f(x,y) dxdy \tag{2.3}$$

are invariants, where $\bar{x} = m_{1,0}/m_{0,0}$, $\bar{y} = m_{0,1}/m_{0,0}$ are the coordinated of the center of mass. The major axis orientation of the region is given by:

$$\theta = \frac{1}{2} arctan \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \tag{2.4}$$

The eccentricity of the region is given by:

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2} \tag{2.5}$$

2. *Scaling* Under a scale change, $\acute{x} = \alpha x$, $\acute{y} = \alpha y$, the moments of $f(\alpha x, \alpha y)$ change to $\acute{\mu} = \mu_{p,q}/\alpha^{p+q+2}$. The normalized moments, defined as:

$$\eta_{p,q} = \frac{\mu^{\alpha}_{p,q}}{\mu_{0,0}}, \quad \alpha = (p+q+2)/2 \tag{2.6}$$

are then invariant to size change. This normalization formula applies only to image regions (not boundaries). The normalized moments which are invarinat to scaling for curves (image boundaires) are defined as:

$$\eta_{p,q} = \frac{\mu^{\alpha}_{p,q}}{\mu_{0,0}}, \quad \alpha = (p+q+\mathrm{I}) \tag{2.7}$$

Note that the magnitude of normalized moments may decrease exponentially with increasing order. This problem can be addressed by using moments defined as:

$$\acute{\eta}_{p,q} = (\eta_{p,q})^{\frac{1}{p+q}} \tag{2.8}$$

3. *Rotation and reflection.* Under a linear coordinate transformation:

$$\begin{bmatrix} \acute{x} \\ \acute{y} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.9}$$

the moment generating function will change. It is possible to find certain polynomials of $\mu_{p,q}$ that remain unchanged under the transformation of the eauation above.

The first seven normalized geometric moments which are invariant under translation, rotation and scaling are given by Hu [28]:

$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + 3(\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{12} - \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - \eta_{12})(\eta_{30} - \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
\phi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\quad +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{03} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad -(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned}
\tag{2.10}
$$

The advantage of using geometric moments descriptors is that it is a very compact shape representation and the computation is low, however, it is difficult to obtain higher order moment invariants.

2.2 Boundary-based methods

2.2.1 Fourier Descriptors

Fourier descriptors are complex coefficients of the Fourier series expansion of waveforms[29]. Given a shape in the Cartesian plane, the boundary points are re-sampled to obtain an l-point closed boundary. l is usually set to 64 or 128. Let $[x_k, y_k]$, k = 0,1,....,l-1, be the coordinates of l samples on the boundary of an image region. For each pair $[x_k, y_k]$ we define the complex variable:

$$u_k = x_k + iy_k \tag{2.11}$$

For the l u_k points we obtain the Discrete Fourier Transform (DFT) f_l:

$$f_l = \sum_{k=0}^{N-q} u_k exp(-i\frac{2\pi}{N}lk), l = 0, 1, ..., N-1 \tag{2.12}$$

The coefficients f_l are known as the *Fourier descriptors* of the boundary, Usually, a small number of coefficients are used. The Fourier descriptors are invariant to the starting point of sampling, rotation, scaling and reflection. Apart form the first coefficient (which gives the centroid of the object) all Fourier descriptors are translation invariant [44].

2.2.2 Chain Codes

Chain code describes an object by a sequence of unit-size line segments with a given orientation [30]. The method was introduced in 1961 by Freeman [31] who described a method permitting the encoding of arbitrary geometric configurations. In this approach, an arbitrary curve is represented by a sequence of small vectors of unit length and a limited set of possible directions, thus termed the unit-vector method. On the grid, encoding is based on the fact that successive contour points are adjacent to each other. Depending on whether the 4-connected or the 8-connected grid is employed, the chain code is defined as the digits from 0 to 3 or 0 to 7, assigned to the 4 or 8 neighboring grid points in a counter-clockwise sense, Fig. 2.2. An illustration is given in Fig. 2.3 . A direct straight-line segment connecting two adjacent grid points is called a link, and a chain is defined as an ordered sequence of links with possible interspersed signal codes [32]. A chain can be coded by the absolute image address of one of its points followed by the relative position of the remaining points to their predecessors, leading to the following bit requirement **B** for a chain of length **n** and an image with size **N x M**:

$$B = ln(N) + lb(M) + (n-1); b(k) \tag{2.13}$$

where **lb(.)** represents the logarithm of base 2 and **k** denotes the connectivity of the contour grid (4 or 8).

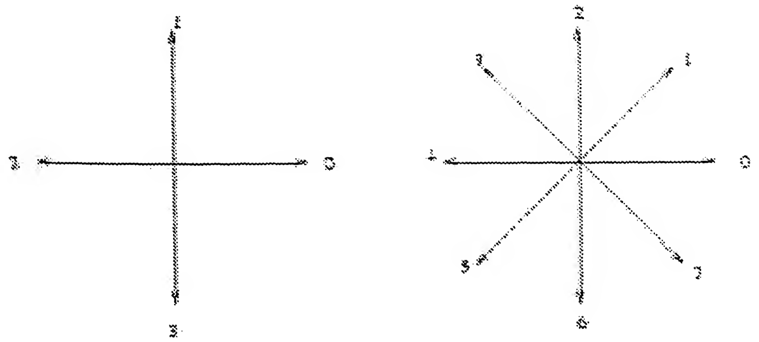


Figure 2.2: Definition of the chain code in the 4-connected and in the 8-connected grid.

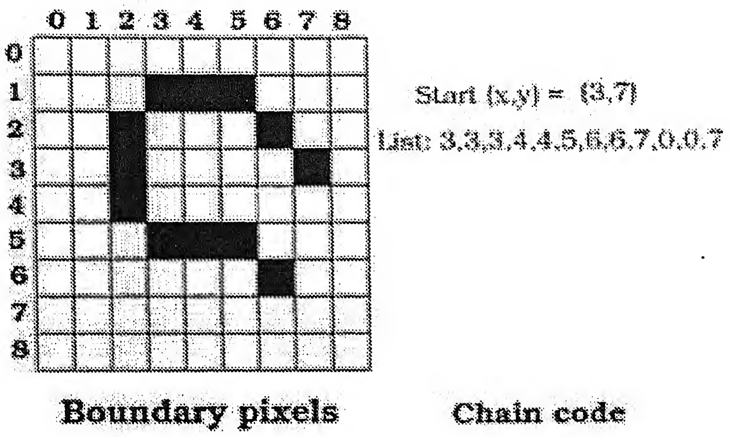


Figure 2.3: The chain code of a boundary segment using 8-connectivity. The boundary is shown on the left. The 8-directional chain code for the example is given on the right.

The chain code usually has high dimensions and is sensitive to noise. It is often used as an input to a higher level analysis. For example, it can be used for polygon approximation and for finding boundary curvature which is a important perceptual feature.

2.2.3 Edge Direction Histogram

Jain and Vailya [1] introduced edge direction histogram (EDH). The edge information from the image was calculated using the Canny edge operator [38] and then the quantization of edge directions into 72 bins, each of five degree, was done. The Euclidean distance metric is used to compute the dissimilarity value between two edge direction histograms. By definition, a histogram directions is invariant to translations in an image. They applied following normalizations to the edge direction histogram:

- *Normalization against scale variations*

  The histogram was normalized with respect to the number of edge points in the image.

- *Normalization against rotation*

  A shift of the histogram bins during the matching partially takes into account a rotation of the image. But, due to the quantization of the edge directions into bins, the effect of rotation is more than a simple shift in the bins. Smoothing of histogram has been proposed by Jain and Vailya, as:

  $$I_s[i] = \frac{\sum_{j=i-k}^{i+k} I[j]}{2k+1} \tag{2.14}$$

  where I_s is the smoothed histogram, I is the original histogram, and the parameter k determines the degree of smoothing.

2.3 Patent Search and Mining

The sheer size of the information available about the patents has led many researchers to apply and develop technologies for information retrieval from patent databases. In [40], Fattori et. al., has developed a system, *PackMOLE*^{TM}, which applies text mining to the patent database. The author says, professional patent searchers are suspicious of the alleged "black box" effect inherently attached to the text mining softwares. They have proposed that to overcome these prejudices, a realistic business objective should be set while experimenting with these tools. On the other hand, there are non-textual informations in the patent databases which need specific techniques for search and retrieval of these informations. Hopkins [41], has described a search for non-word US trademarks using codes from the Design Search Code Manual. The codes are used in an electronic search, either on the USPTO website or on CASSIS DVDs. The application of a such a system is in identifying the conflicting trademarks with different text, but confusingly similar graphics.

Chapter 3

PROPOSED SYSTEM

3.1 The Overview of the System Architecture

The overall architecture of the system can be divided into two sub-systems according to their purposes, which are ***the Database Creation System*** and ***the Database Retrieval System***. Figure 3.1 illustrates the relationships between the two sub-systems and their components.

The Database Creation System creates an image feature database. It provides a user interface where the user can enter keyword and the system searches the United States Patent Office (USPTO) patent database and grabs the images for the patents satisfying the search. These images are then stored in the image database, which is used later for browsing and results display. For each image, features are extracted and the image feature database is populated with them.

The *Database Retrieval System* provides a user interface that accepts a query in terms of an image. The image is processed to extract its features, then the image feature database is searched in order to output the retrieval results.

3.2 The Database Creation Process

As shown in Figure 1.1, a main component of a Content Based Image Retrieval (CBIR) is the visual feature repository. The creation of this repository comes early in the development of a CBIR. The database creation process is depicted in the Figure 3.2. We have developed a user

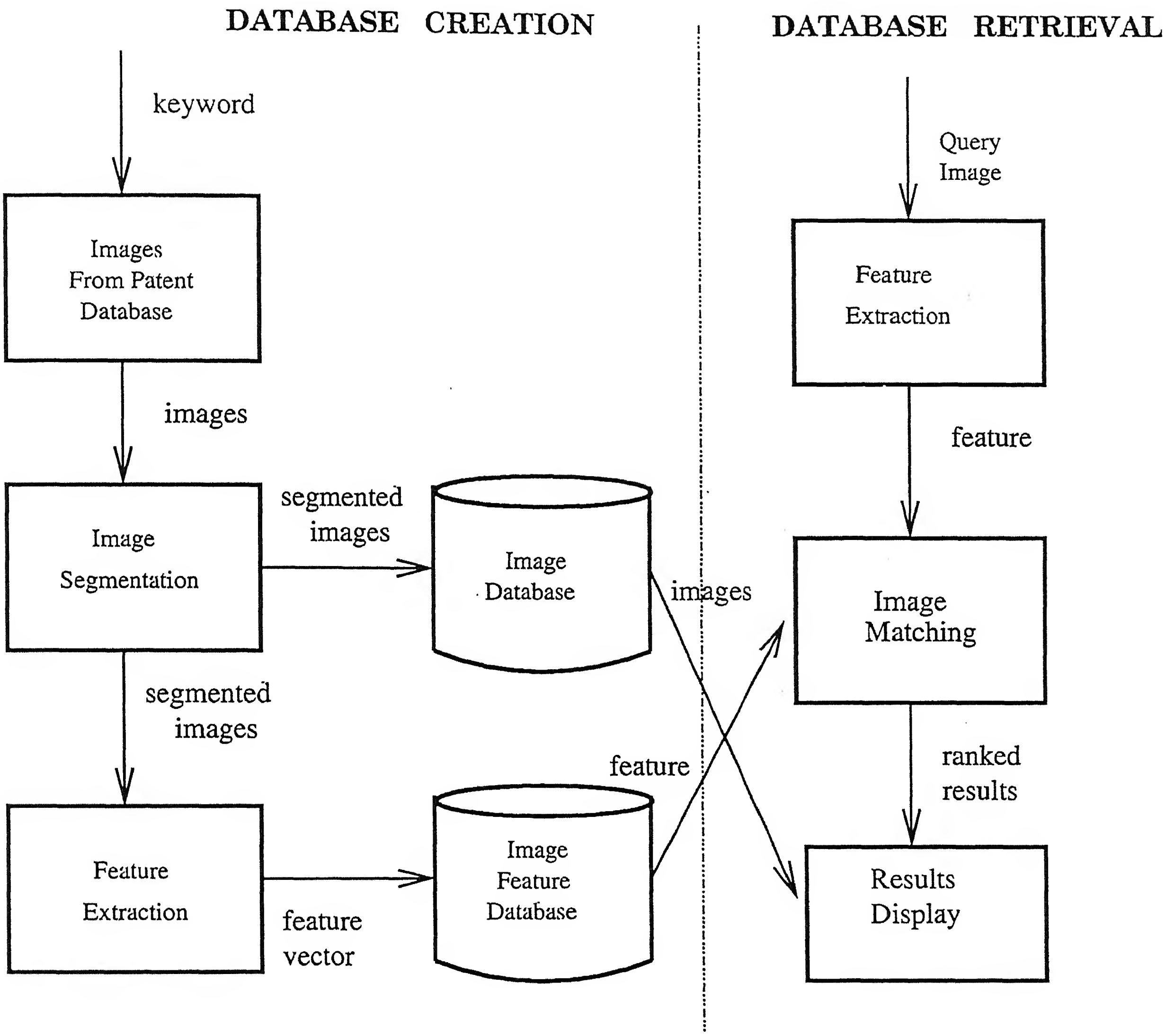


Figure 3.1: Proposed Content Based Image Retrieval System Architecture

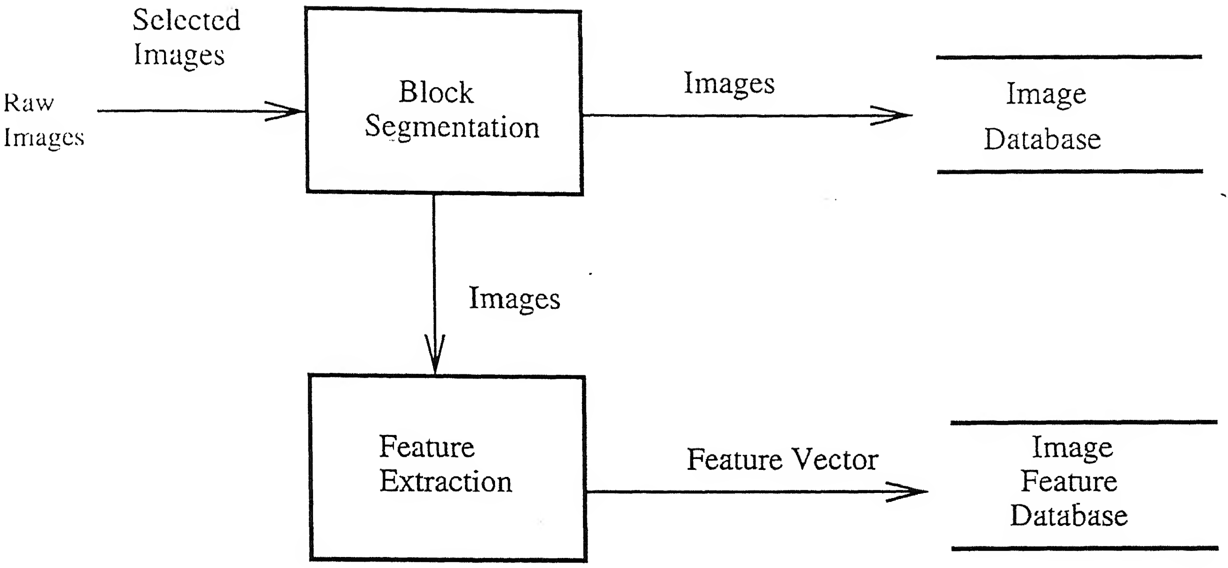


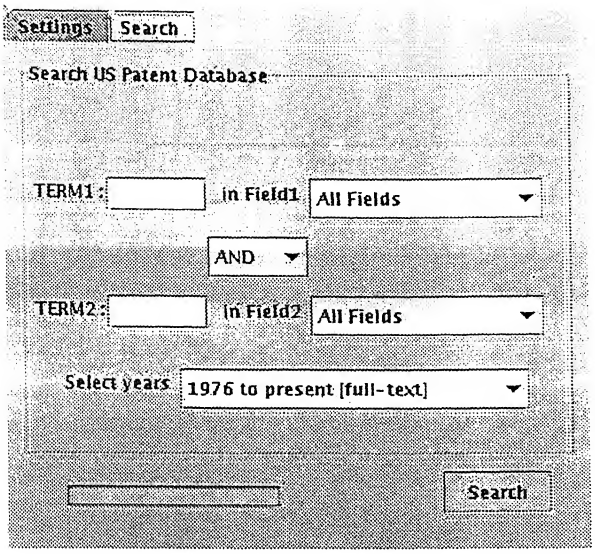
Figure 3.2: The Database Creation Process

interface (Figure 3.3), for specifying the search criteria for the patents. The patents fulfilling the search requirement are then automatically grabbed from the USPTO website http://www.uspto.gov. After obtaining the pages that contain images, the block segmentation differentiate between the graphic and other contents of the page. The graphic content are then used to calculate the image feature vector which is then stored in the database along with patent number and the page number within the patent where this image was found. The process of image block segmentation and feature extraction is described in Section 3.2.1 and 3.2.2.

3.2.1 Image Segmentation

The patents are stored as image documents in the USPTO repository. A typical patent page is shown in the Figure 3.4. To extract the drawings from these pages we need to identify the image and text blocks. Thus we need to differentiate between the portions of document which have graphic and text contents.

Using run-length smoothing algorithm(RSLA) [42], the document image is subdivided into blocks (regions), each of which contain either only text or graphic (possibly with some text) content. The blocks with graphic content are identified and rest are discarded. A document and its



TERM1: in Field1 All Fields

AND

TERM2: in Field2 All Fields

Select years 1976 to present [full-text]

Figure 3.3: Patent Grabber User Interface of PATSEEK

blocks identified after segmentation are depicted in Fig 3.4.

3.2.2 Feature Extraction

An image can be processed to produce numeric descriptors capturing specific visual characteristics called *features*. The important features for image databases are based on color, texture and shape of the image. We have used a *shape-based* feature, edge-orientation auto correlogram (EOAC) . The edge orientation autocorrelogram (EOAC) classifies edges based on their orientations and correlation between neighboring edges [43]. The EOAC has the following properties:

1. It includes the correlation between neighboring edges in a window around the kernel edge
2. It describes the global distribution of local correlation of images
3. It describes shape aspect of an image and thus it is not sensitive to color and illumination variation
4. It acts independent of translation and scaling.

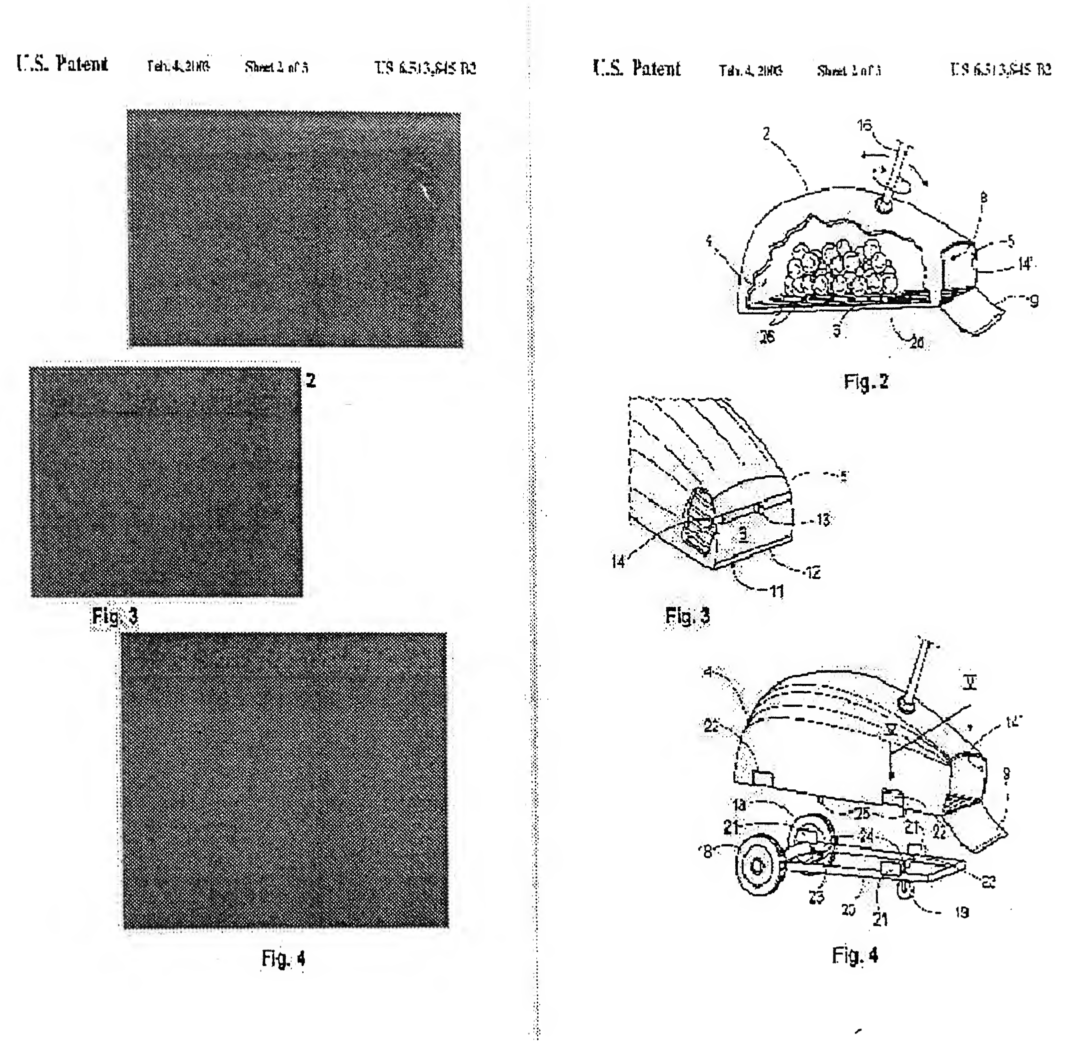

U.S. Patent Feb. 4, 2003 Sheet 2 of 3 US 6,513,845 B2
2
Fig. 3
Fig. 4
U.S. Patent Feb. 4, 2003 Sheet 2 of 3 US 6,513,845 B2
Fig. 2
Fig. 3
Fig. 4

Figure 3.4: Image Segmentation. A patent page is displayed in the right hand side. The image on the left hand shows the result of image block segmentation. The blocks containing graphic contents is covered with gray rectangles.

5. It is easy to compute, and

6. The size of the resulting feature vector is small. Image's EOAC takes only 144 real numbers for storage.

Algorithm

The algorithm for generating EOAC consists of five steps as follows[43]:

(1) *Edge detection*: Edges form the outline of an object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects. This means that if the edges in an image can be identified accurately, all of the objects can be located and basic properties such as area, perimeter, and shape can be measured. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

An example of edge detection is illustrated in Figure 3.5. There are two overlapping objects in the original picture (3.5 (a)), which has a uniform gray background. The edge enhanced version of the same image (3.5 (b)) has dark lines outlining the three objects. Note that there is no way to tell which parts of the image are background and which are object; only the boundaries between the regions are identified. However, given that the blobs in the image are the regions, it can be determined that the blob numbered 3 ovelaps with blob 2 . This information can be used to analyze the image further.

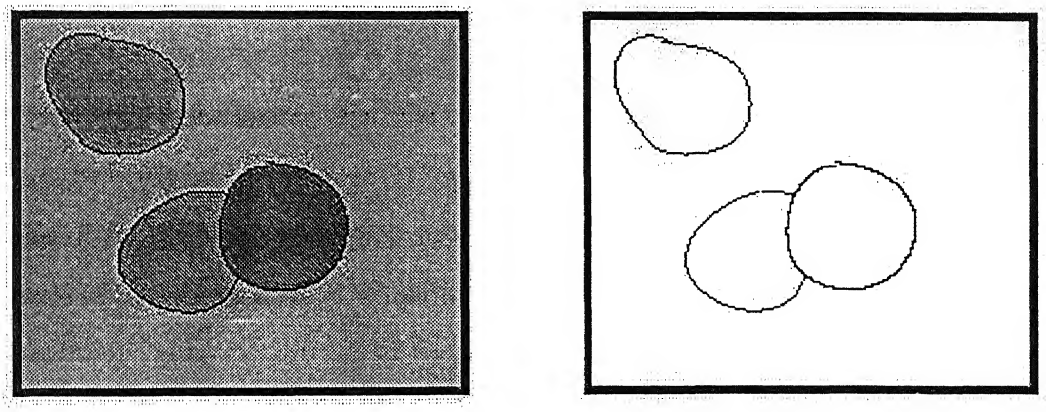


Figure 3.5: Example of edge detection. (a) Synthetic image with blobs on a gray background. (b) Edge enhanced image showing only the outlines of the objects.

Technically, edge detection is the process of locating the edge pixels, and edge enhancement increases the contrast between the edges and the background so that the edges become more visible. In practice these terms are used interchangeably, since most edge detection programs also set the edge pixel values to a specific gray level or color so that they can be easily seen. In addition, edge tracing is the process of following the edges, usually collecting the edge pixels into a list. This is done in a consistent direction, either clockwise or counter-clockwise around the objects. Chain coding is one example of a method of edge tracing. The result is a non-raster representation of the objects which can be used to compute shape measures or otherwise identify or classify the object.

Gradient operators

The gradient of a digitized image *f(x, y)* is defined as:

$$\mathbf{G}[f(x,y)] = \begin{bmatrix} G_x \\ \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \\ \frac{\delta f}{\delta y} \end{bmatrix} \tag{3.1}$$

It is well known from vector analysis that the vector **G** points in the direction of maximum rate of change of f at location (x, y). For edge detection, we are interested in the magnitude of this vector, generally referred to as the *gradient* and denoted by G[f(x, y)], where

$$G[f(x,y)] = [G_x^2 + G_y^2]^{1/2}. \tag{3.2}$$

This quantity is equal to the maximum rate of change of f(x, y) per unit distance in the direction of G.

The *direction* of the gradient vector is also an important quantity. Letting α(x, y) represent the direction angle of **G** at location (x, y), it follows from vector analysis that

$$\alpha(x,y) = tan^{-1}(G_y/G_x), \tag{3.3}$$

where the angle is measured with respect to the x axis.

Template Based Edge Detection

From Eq. 3.1, computation of the gradient is based on obtaining the partial derivatives $\delta f/\delta x$ and $\delta f/\delta y$ at every pixel location. One way is to convolve an image f(x, y) with the *Sobel operators* given below. The responses of these two operators at any point (x, y) are combined using Eq. 3.2 to obtain the gradient at that point.

$$S_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} S_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.4}$$

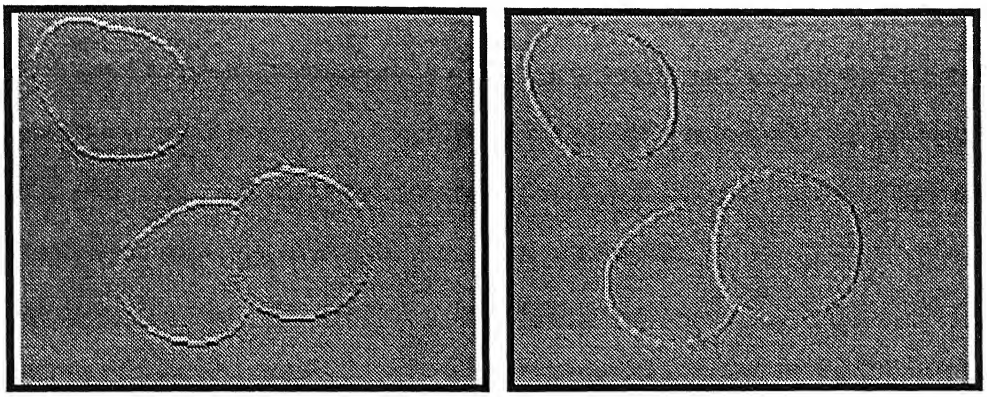


Figure 3.6: Results of applying (a) Horizontal Sobel Operator S_X and (b) Vertical Sobel Operator S_y to image shown in Fig 3.5

The Sobel operator is less sensitive to noise than other edge detectors [44]. Therefore it has been used for edge detection and making the gradient image.

(2) *Finding prominent edges*: This step extracts the prominent edges of the gradient image. The prominent edges are extracted by comparing all the edge amplitudes with a threshold value T_1. We have chosen $T_1 = 25$, which is approximately 10 percent of the maximum intensity value in the 8-bit original images .

(3) *Edge orientation quantization*: This step quantizes edges uniformly into n segments $\angle G_1$; $\angle G_2$; $\angle G_3,...,\angle G_n$ and each segment is equal to five degrees.

(4) *Determining distance set*: This step constructs a distance set (D), which shows the dis-

tances from the current edge that is used in calculating correlation. It is clear that near edges have high correlation together, thus the number and value of members of D must be low. In our work, we have chosen a set with four members as shown below:

$$D = 1,3,5,7 \qquad and \qquad d = |D| = 4 \tag{3.5}$$

There is no need to consider the pixels with even numbers, because most of their information is in their adjacent pixels with odd numbers. For example, the correlation information associated with 2 pixels apart can be extracted from 1 and 3 pixels apart.

(5) *Computing elements of EOAC*: In the final stage, the edge orientation autocorrelogram is constructed. This correlogram is a two-dimensional array (a matrix), consisting of n rows and d columns. The ⟨j, k⟩ element of this matrix $(1 \leq j \leq n, k \in D)$ indicates the number of similar edges with the orientation $\angle G_j$, which are k pixel distance apart. Two edges with k pixel distance apart are said to be similar if the absolute values of their orientations and amplitude differences are less than an angle and an amplitude threshold value, respectively [44].

Normalization

Humans has the ability to recognize similar images irrespective of the following five factors: translation, rotation, scaling, color, and illumination variations. For a CBIR to be effective, it should be invariant of these factors. In PATSEEK, we taken following steps for making for making it invariant to these factors:

1. *Normalization against translation*

   EOAC is naturally translation invariant, because translation has no effect on amplitude and orientation of edges.

2. *Normalization against scaling transform*

   Since the total number of edge pixels depends to the total number of image pixels, image scaling affects the total number of edges. In contrast, image scaling has no effect on their orientation and amplitude, because edges are constructed on the borders of regions with different colors when an image is resized its regions relative position color remain unchanged. For these reasons, the total number of edges is scaled uniformly in the EOAC surfaces. The feature vector normalization is made invariant by dividing the number of edges is each bin by the total number of edges in the image.

3. *Color and Illuminance Invariance*

   The images stored in patent databases are according to international standards. The images are bi-level TIFF format images. Thus the images are already invariant to color and illumination variations, and we need not take any measures for normalizing the image feature vector against them.

4. *Handling rotation transforms*

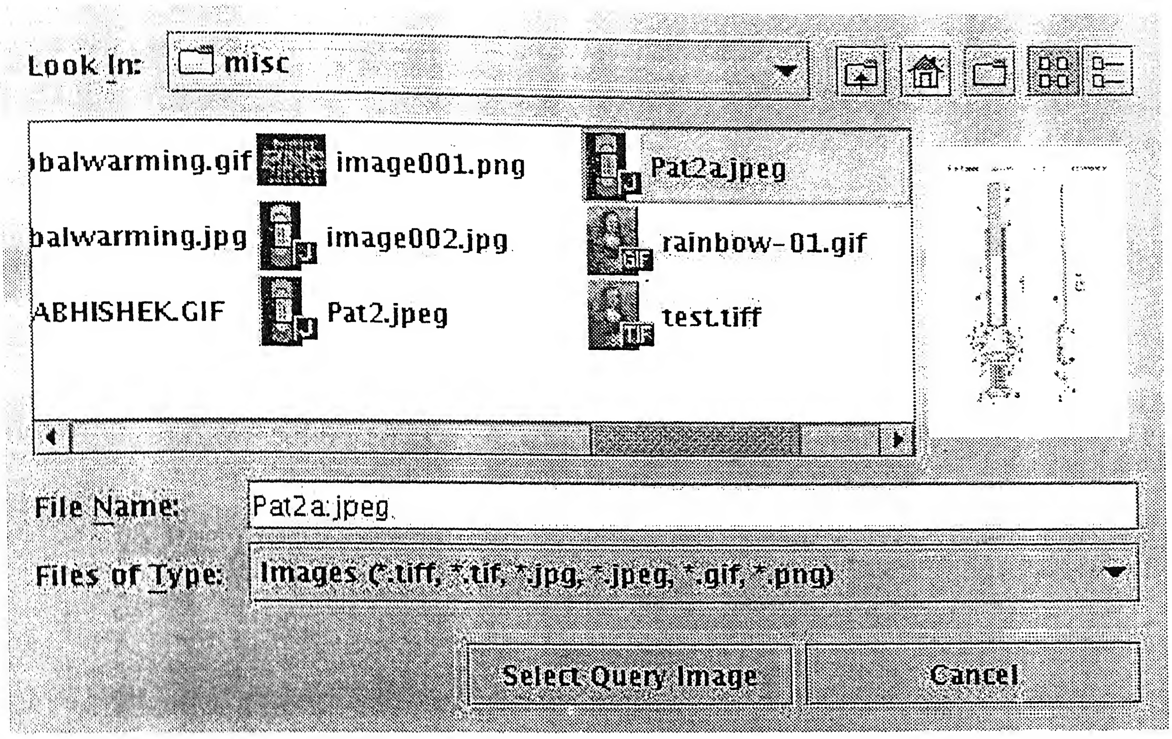
   PATSEEK asks the user to specify the rotation angle for the query image while image retrieval. Thus, if the user wants to check for cosmetic changes in the query image. Thus we have not made changes to image feature representation for rotation invariance.

3.2.3 Image Feature Database

The image features calculated a priori are stored in a "*tabular*" structure, which is supported by database management systems (DBMSs). Such design faciliates use of DBMS. We use SQL (Structured Query Language) to retrieve the image feature.

3.3 The Database Retrieval System

The database retrieval system provides an interactive interface to the user for selecting the query image from the image database (refer Figure 3.7). The interactive interface consists of a file browser and is designed to automatically display the thumbnails of the selected image. The feature vector for the selected query image is calculated as already explained in the section 3.2.2. This feature vector is then compared with every feature vector stored in the image feature database. The



Look In: misc

File Name: Pat2a.jpeg

Files of Type: Images (*.tiff, *.tif, *.jpg, *.jpeg, *.gif, *.png)

Figure 3.7: Query Image Selection

top 12 results are then displayed to the user. Image matching and user interface will be described in Section 3.3.1 and 3.3.2.

3.3.1 Image Matching

The feature vector of the query image is matched to every image feature vector stored in the image feature database. As the image feature database is implemented on a DBMS, the SQL queries are used to extract the feature vectors from the database. For each feature vector, it's distance is calculated with respect to the query vector. The top twelve images, ranked on the basis of the distance are displayed as thumbnails along with the respective distance to the user. The thumbnails are sorted in the increasing order of the distance with the respective image feature vector has with the query image feature vector. L1 distance was employed as the similarity/dis-similarity measure for matching images. If $X = [x_1, x_2, ..., x_n]$ and $Y = [y_1, y_2, ..., y_n]$ are two featuer vectors then the L1 distance metric between X, Y is given by:

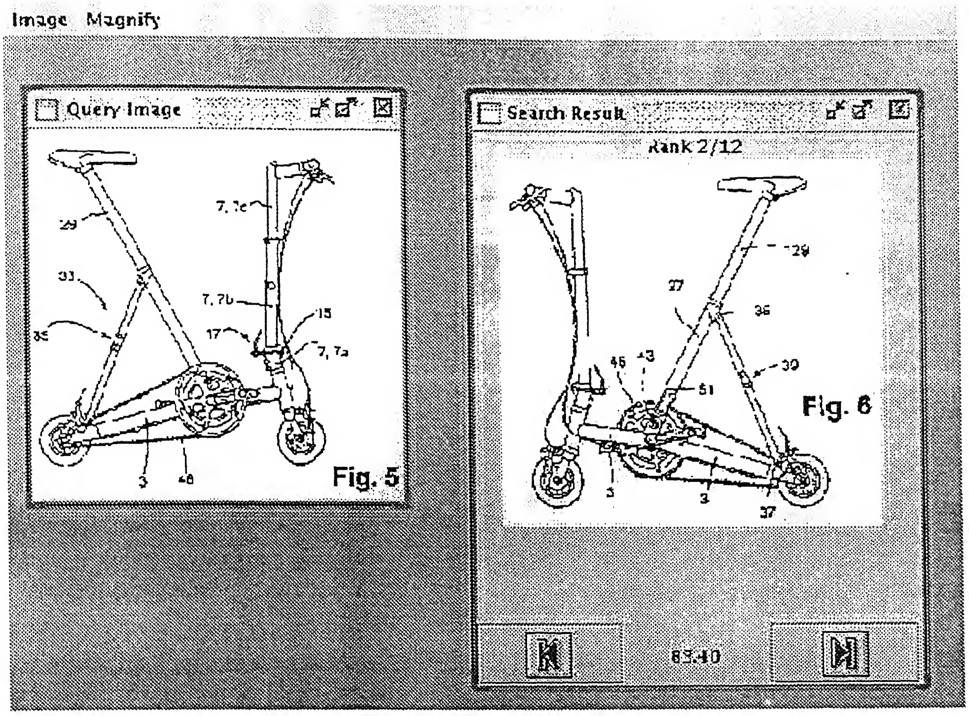


Figure 3.8: The User Interface for query result navigation

$$L1(X,Y) = \sum_{i=1}^{n} |x_i - y_i|$$

3.3.2 User Interface

The image matching process gives a list of top twelve images which are nearest to the query image according to the chosen similarity measure. The user interface has a graphical interface which displays the query image and the results for browsing to the user. A snapshot of the user interface is shown in the Figure 3.8. The design of the user interface was inspired by the common facility in current image retrieval systems which provide a thumbnail based method to display the retrieved images. In our case, due to the shear size of the ranked images, same cannot be properly displayed together. In order to overcome this limitation, the ranked images are displayed one by one and a browsing facility has been provided.

Chapter 4

EXPERIMENTS

In this chapter, we report an experimental study conducted to study the effectiveness of the prototype system developed.

4.1 Set Up

The prototype system consists of an image database, a set of benchmark queries, a set of relevant images, and a set of evaluation metrics. All experiments were performed on an Intel Pentium IV Processor 2.4 GHz with 512 MBytes of RAM. The proposed system was implemented in Java language(Sun JDK 1.4.1). For implementing the image feature database MySQL Ver 12.21 was used as RDBMS. We evaluated the two similarity measures L1 and L2 distance on their effectiveness.

- **Image Collection** The Content Based Image Retrieval community lacks the availability of a standard image collection for performance evaluation. Thus we have developed our own collection for the proposed system. Our collection consists of around 700 images from the patent database of United States Patent Office (http://www.uspto.gov). All the images in our collection are gray images. Table 4.1 list the patents whose images are used in our image collection. Figure 4.1 show five images randomly picked out from the collection.

6553641	6557265	6560876	6560881
6571476	6581290	6584696	6588113
6598303	6607324	6615498	6616658
6629475	6634108	6634492	6651342
6655029	6675479	6684513	6691415
6694618	6694626	6701619	6702495
6705789	6711528	6722039	6725549
6729785	6733402	6739054	6749788
6568082	6568084	6578266	6722803
6725550	6698100	6708408	6598301
6604517	6623384	6494340	6688626
6694846	6736450	6739664	6708584
6739465	6746205	6746301	6425835
6432004	6461259	6645094	6648780
6695184	6695334	6712723	6746300
6749265	6435991	6447411	6634548
6645096	6712371	6691691	6747225

Table 4.1: List of United States patent numbers used for making image collection

- **Benchmark Queries and relevant Images** For the performance evaluation, we have chosen eighteen images from our collection. For each query image, a set of relevant images have been identified. The relevant images are very similar to their query image with some differences on scaling, translation, and viewing position variation. Ideally, when a query is performed all of its relevant images should be retrieved in lower ranks.

- **Performance evaluation metrics** We have analyzed the performance in terms of *retrieval accuracy*. This term is concerned with effectiveness of image retrieval. For this purpose, many researchers have computed *precision* and *recall rates* as two accuracy metrics. Muller et al. [39] has defined them as:

$$precision = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ images\ retrieved} \tag{4.1}$$

$$recall = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ number\ of\ relevant\ images\ in\ the\ collection} \tag{4.2}$$

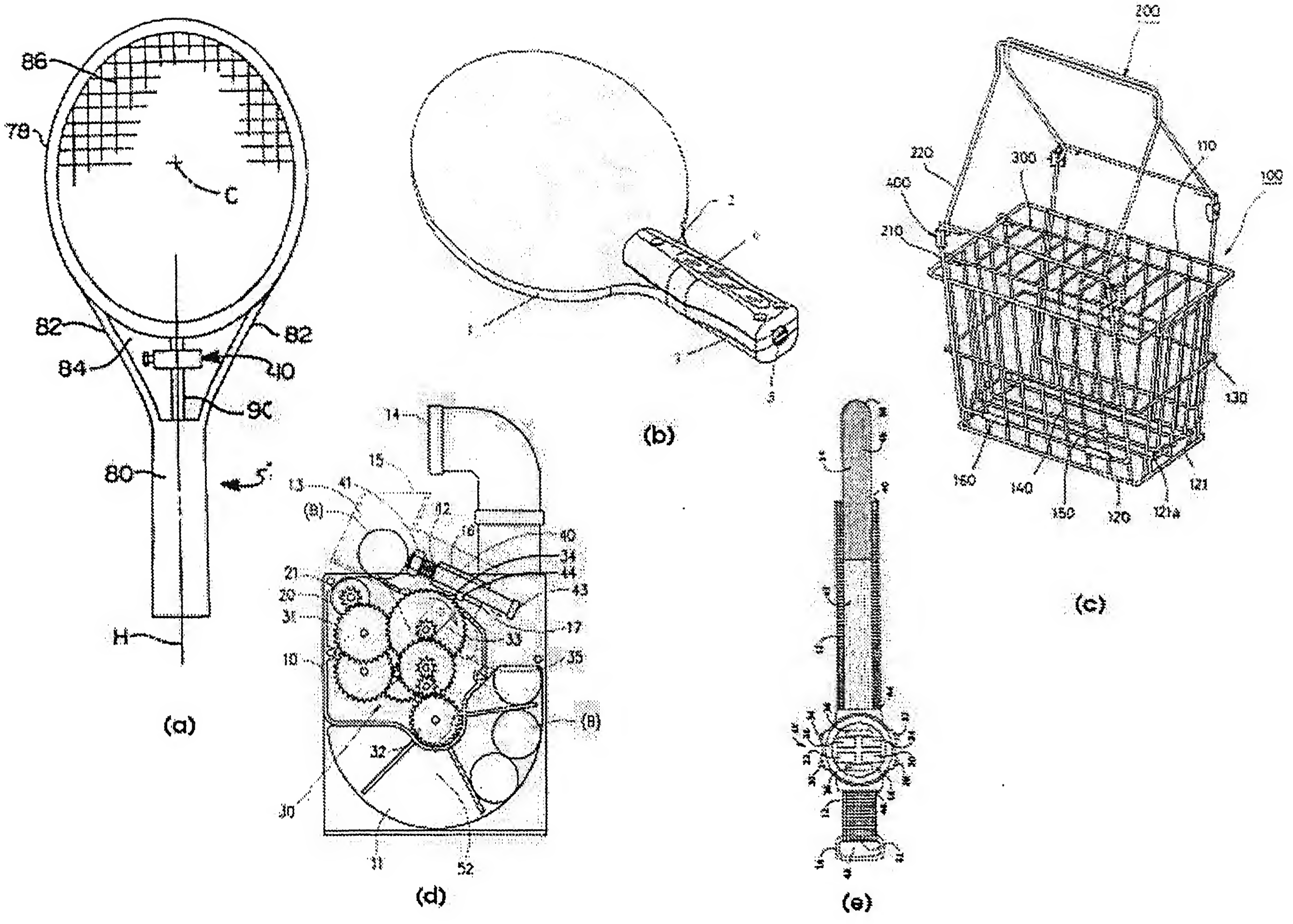


Figure 4.1: A sample of images from our image collection (a)-(e).

4.2 Performance Evaluation

An efficient content based image retrieval system should have following features:

- *Accuracy* The retrieval system must be accurate. i.e., the retrieved images must resemble the query image. We classify a retrieval as accurate if for a given query image, it's relevant images in the database are retrieved by the system in the top twelve results. We have compared the performance of two similarity measures, L1 and L2 distance, in our experiments.
- *Speed* It is desirable to have an efficient retrieval system. Since image databases typically have thousands of images, the retrieval scheme must be "real-time". The total time taken for a retrieval on the entire database was measured in our experiments.

Accuracy of the System A set of eighteen images were selected as benchmark queries and the relevant images were decided for each benchmark query. For each image precision and recall rate was calculated. Figure 4.2 and 4.3 shows the precision and recall rate of the system using L1 and L2 distance metrics as similarity measure for each benchmark query image. As evident from the graphs, the performance of L1 and L2 distance is almost similar except for one query image.

Speed The query by image example was evaluated. A query image was given, it's feature vector was extracted from it and then the database was queried for features vectors of images already stored. The query vector's L1 distance with every stored feature vector is calculated and the top twelve results are stored for display. The system took around 1 min 30 seconds for this operations.

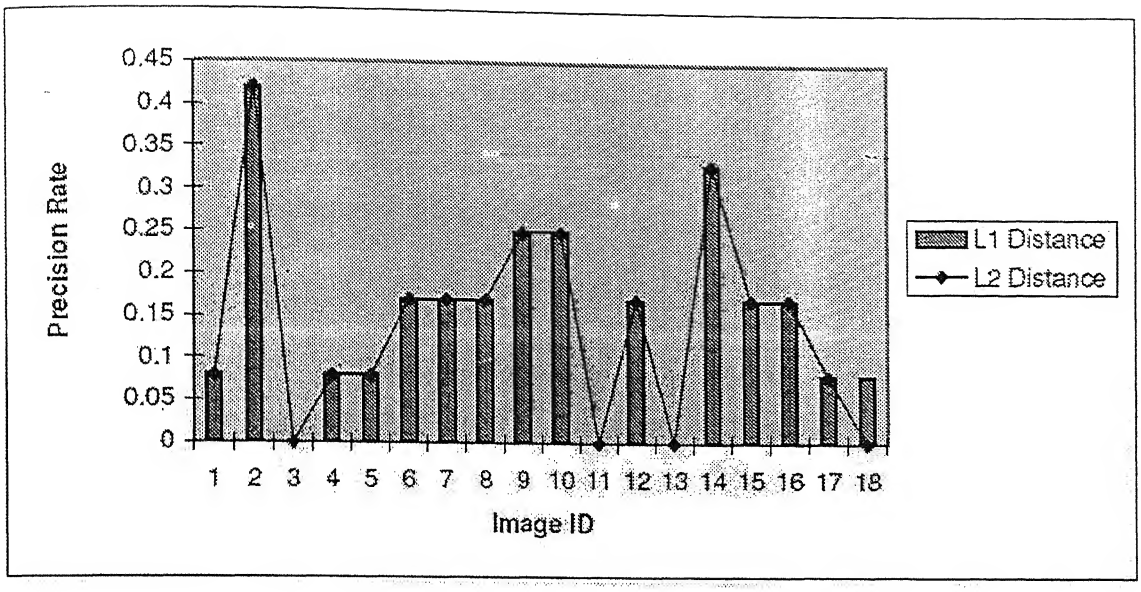


Figure 4.2: Precision rate of the system using L1 and L2 distance mertrics as similiarity measure

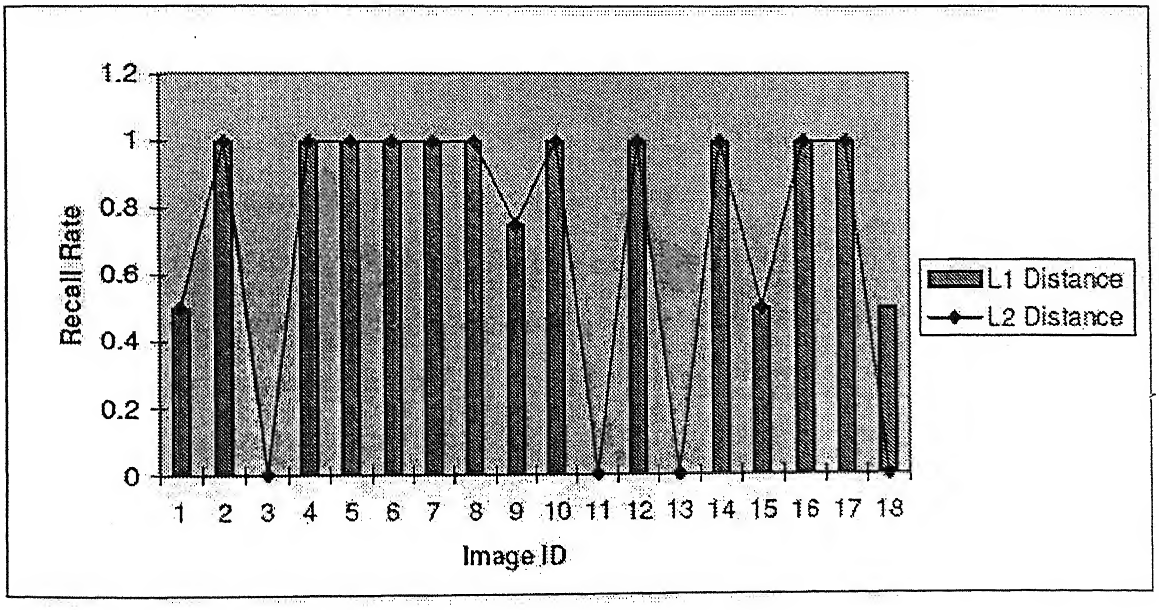


Figure 4.3: Recall rate of the system using L1 and L2 distance mertrics as similiarity measure

Chapter 5

CONCLUSION AND SCOPE FOR FUTURE WORK

The professional patent searchers in the past have restricted themselves to the implementation of text based search to search the patent database. The patent database have two main components, namely, text and the image. In this thesis, we develop an image retrieval system capable of automatically creating an image database and content based search on the created database. The United Stated Patent Office (USPTO) patent database has been used as the parent patent database. Based on the keywords supplied by the user, the system searches the USPTO and retrieves the full page patent images. These full page patent images are then locally processed to separate the drawings. The drawings are then stored in the local database.

The other part of the system developed, provides a user interface to search the local database using query-by-image. The user selects an input image and the system provides as output top twelve images which are most similar to the query image in the database.

The combination of the text and image based search techniques can be an effective tool in the hands of the professional patent searchers. The combined technique could pave way for systems that would have high accuracy than the present text based or the image based search systems.

Bibliography

[1] A.K. Jain and A. Vailaya, "Image Retrieval using Color and Shape", *Pattern Recognition*, Vol. 29,pp.1233-1244,August,1996.

[2] T. Kato, "Database architecture for content-based image retrieval," in Image Storage and Retrieval Systems (Jambardino A and Niblack W eds), Proc SPIE 2185, pp112-123, 1992.

[3] Y.Rui, T.S.Hang, and S.-Fu Chang, "Image Retrieval: Current technique, promisinsg directions, and open issues," *J.Vis. Commun. Image Represent.* **10**, 39-62 (1999)

[4] M.Flickener et al., "Query by image and video content: the QBIC system", *IEEE Comput.* **28**, 23-32 (1995)

[5] J.R.Bach et al., "The virage image search engine: an open framework for image management", *Proc. SPIE: Storage and retrieval for Still Image and Video Databases IV*, **2670**, 76-87 (1996)

[6] R. Elmasri and S.Navathe, *Fundamentals of Database Systems*, Benjamin Cummings, Addision Wesley, Boston, MS, 1994 pp. 116-128.

[7] Y.Rui, T.S.Huang, ans S.Mehrotta, "Content-based image retrieval with relevance feedback in MARS," *Proceedings of International Conference on Image Processing,* Vol. 2,pp. 815-815, 1997.

[8] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying,"*IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 8, pp. 1026-1038, 2002.

[9] H.Voorboes, and T.Poggio, "Computing texture boundaries from images'" *Nature*, 333:364-367, 1988.

[10] M. J. Swain, and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp.11-32, 1991.

[11] J.Hafner, *et al.*, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7,pp. 729-736, July 1995.

[12] W.Niblack *et al.*, "Quering images by content, using color, texture, and shape'"*SPIE Conference on Storage and Retrieval for Image and Video Database*, Vol. 1908, pp. 173-187, April 1993.

[13] T.P.Minka, and R.W.Picard, "Integration learning using a 'society of models'," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 447-452, 1996

[14] W. J. Krzanowski, *Recent Advances in Descriptive Multivariate Analysis*, Chapter 2, Oxford science publications, 1995.

[15] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system." *IEEE Computer*, Vol.28, No.9, pp. 23-32, Sept. 1995.

[16] J.A. Catalan, and J.S. Jin, "Dimension reduction of texture features for image retrieval using hybrid associative neural networks," *IEEE International Conference on Multimedia and Expo*, Vol.2, pp. 1211 -1214, 2000.

[17] N. Beckmann, et al, "The R*-tree: An efficient robust access method for points and rectangles," *ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City, May 1990.

[18] J. Vendrig, M. Worring, and A. W. M. Smeulders, "Filter image browsing: exploiting interaction in retrieval,"*Proc. Viusl'99: Information and Information System*, 1999.

[19] J. T. Robinson, "The k-d-B-tree: a search structure for large multidimensional dynamic indexes," *Proc. of SIGMOD Conference*, Ann Arbor, April 1981.

[20] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: an adaptable symmetric multikey file structure,"*ACM Trans. on Database Systems*, pp. 38-71, March 1984.

[21] H. J. Zhang, and D. Zhong, "A Scheme for visual feature-based image indexing,"*Proc. of SPIE conf. on Storage and Retrieval for Image and Video Databases III*, pp. 36-46, San Jose, Feb. 1995.

[22] *Image Databases*, John Wiley & Sons, Inc.,New York, US (2002)

[23] http://www.uspto.gov/patft/help/datesdb.htm.

[24] S. Livens, P. Scheunders, G. Van de Wouwer, asn D. Van Dyck, "Wavelets for Texture Analysis",*Proceedings of the 6th IEEE International Conference* pp. 814-816, 1997.

[25] Amanda Buckingham, Edge Detection and Image Retrieval Techniques in the interpretation of magnetic images for mineral exploration,*PhD thesis*, The Unversity of Western Australia, 2003

[26] http://algolist.manual.ru/graphics/quant/qoverview.php.

[27] http://www.cs.wpi.edu/ matt/courses/cs563/talks/color_quant/CQindex.html

[28] Hu, M. K. Visual pattern recognition by moments invariants. *IRE Transactions on Information Theory* **IT-8**, pp. 179-187, 1962

[29] Persoon, E. and Fu, K. S. "Shape discrimination using Fourier descriptors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8,pp.388-397,1986.

[30] Zhang D. and Lu G. "Review of shape representation and description techniques". *Pattern Recognition*, 37(1):1-19, 2004.

[31] Freeman H. “On the encoding of arbitrary geometric configurations”.*IRE Trans. on Electronic Computers*, Vol. EC-10, pp. 260–268, 1961

[32] Freeman H. "Computer processing of line-drawing images". *ACM Computing Surveys*, Vol. 6, No. 1, pp. 57–97, March 1974.

[33] A. Pentland, R. W. Picard, and S. Sclaroff. "Photobook: Content-based manipulation of image databases". *International Journal of Computer Vision*, 18(3):233 254, 1996.

[34] M. Beigi, A. Benitez, and S.-F. Chang." MetaSEEk: A content-based meta search engine for images".*In Storage and Retrieval for Image and Video Databases*, SPIE Proceedings Series, San Jose, CA, 1998.

[35] J. R. Smith and S.-F. Chang." VisualSEEk: a fully automated content-based image query system".*In Proceedings of the ACM Multimedia '96*, November 1996.

[36] W. Y. Ma and B. S. Manjunath." NETRA: A toolbox for navigating large image databases".*In Proceedings of IEEE International Conference on Image Processing*, volume I, pages 925 928, Santa Barbara, California, October 1997.

[37] T. S. Huang, S. Mehratra, and K. Ramchandran." Multimedia analysis and retrieval system (MARS) project".*In Proceedings of the 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*. University of Illinois at Urbana-Champaign, March 1996.

[38] J. Canny," A computational approach to edge detection", *IEEE trans. Pattern Anal. Mach. Intell.* PAMI-8, 679-698 (1986).

[39] H. Muller, w. Mullerm D. McG. Squire, S. Marchand-Maillet, and t. Pun. "Performace evaluation in content-based image retrieval: overview and proposals".*Pattern Recognition Letters*. Vol. 22 593-601 (2001).

[40] M. Fattori, G. Pedrazzi and R. Turra,"Text mining applied to patent mapping: a practical businedd case",*world Patent Information*, 25 335-342, 2003.

[41] D. K. Hopkins, "Searching for graphic content in USPTO trademark databases",*World Patent Information*, 25 107-116, 2003.

[42] L.Abele, F. Wahl, and W. Scheel, "Procedures for an Automatic Segmentation of Text, Graphic and Halftone Regions in Documents", *Proceddings of the 2nd Scandinavian Conference on Image Analysis*, Halsinki, 1981.

[43] F. Mahmoudi, J. Shanbehzadeh, A.-M. Eftekhari, and H. Soltanian-Zadeh, "Image retrieval based on shape similarity by edge orientation autocorrelogram",*Pattern Recognition* 36,pp.1725-1736, 2003.

[44] R.C. Gonzalez, P. Wints,*Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.